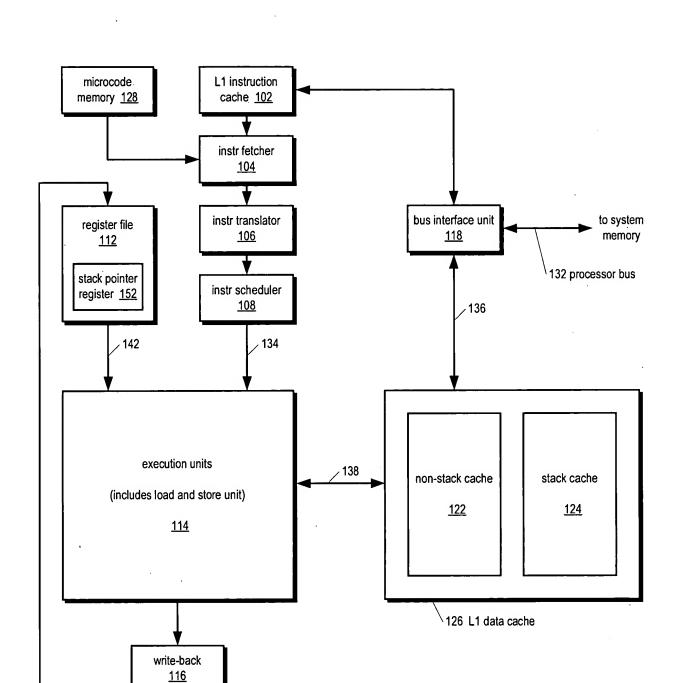
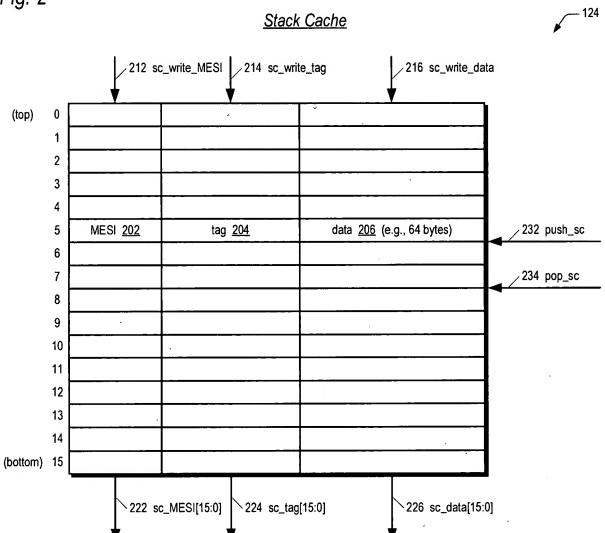
Fig. 1



Microprocessor







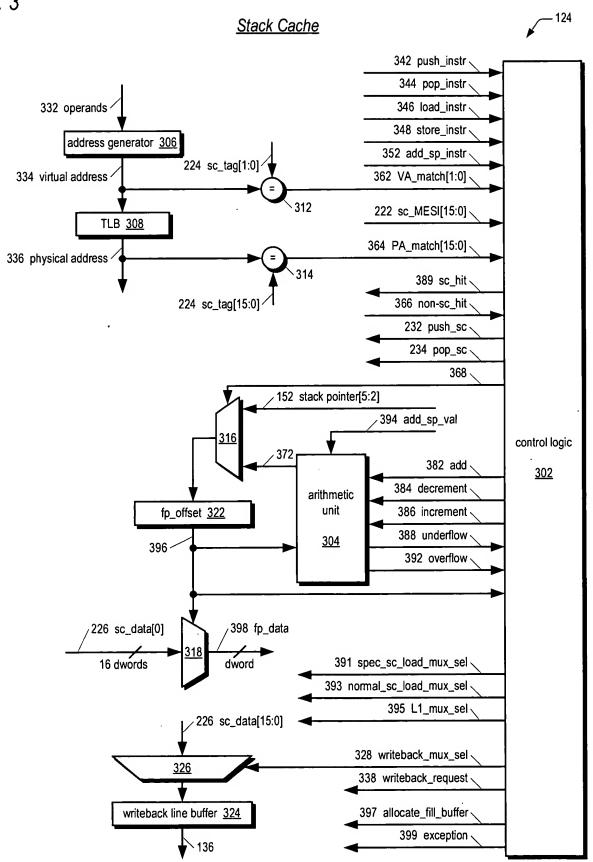




Fig. 4

L1 Data Cache



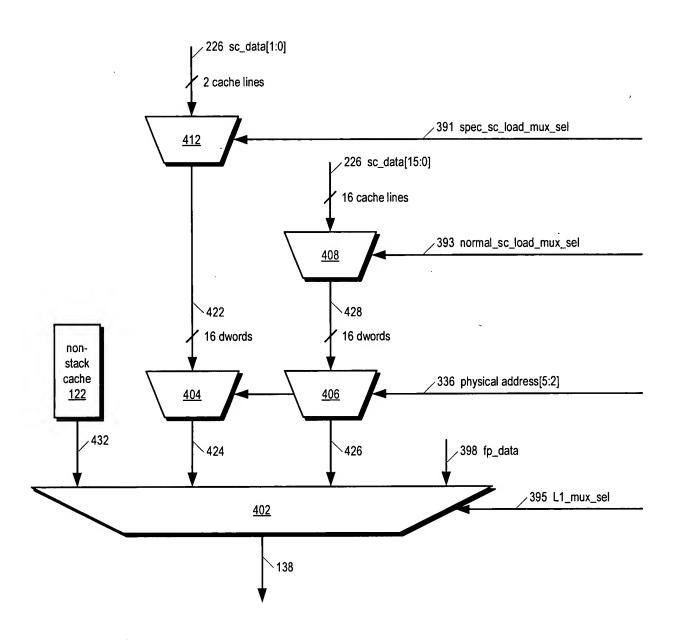


Fig. 5

Fast Pop Operation

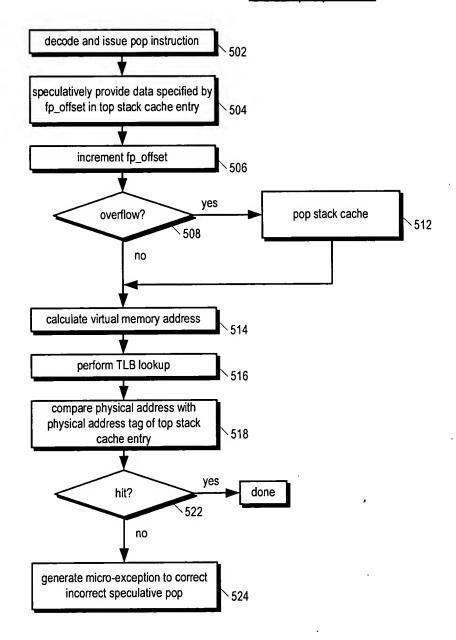


Fig. 6

Push Operation

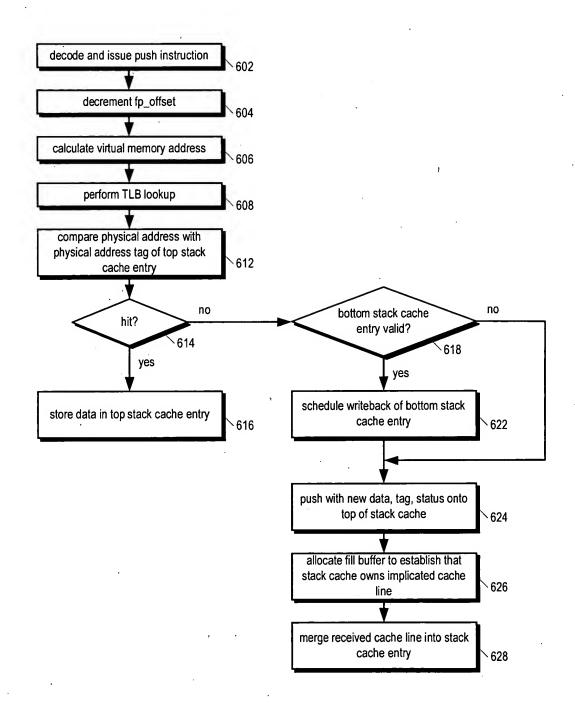


Fig. 7

Add to Stack Pointer Operation

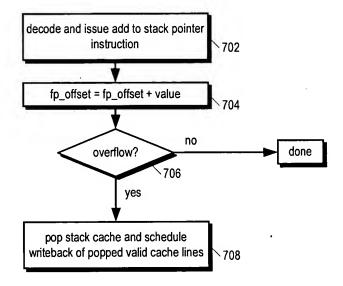
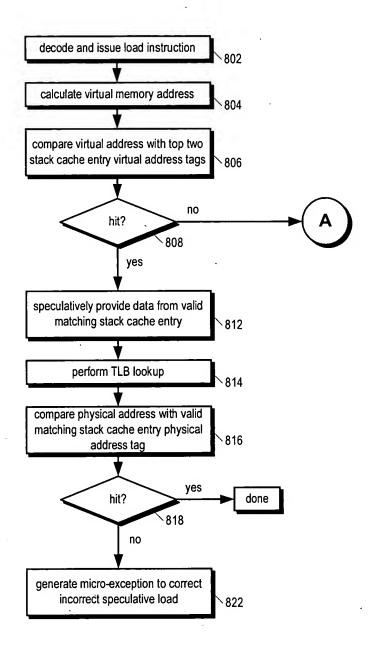


Fig. 8A

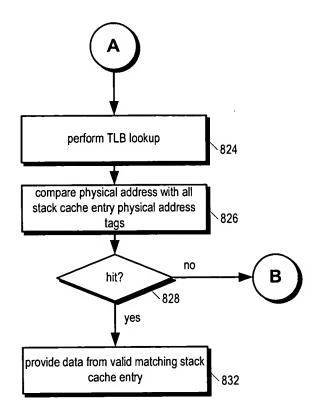
Speculative Load from Stack Cache Operation



4

Fig. 8B

Normal Load from Stack Cache Operation

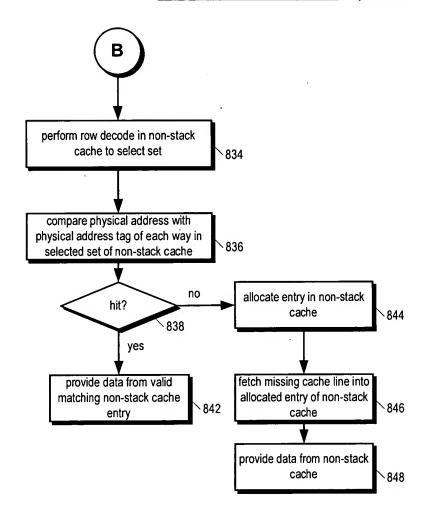


-

+

Fig. 8C

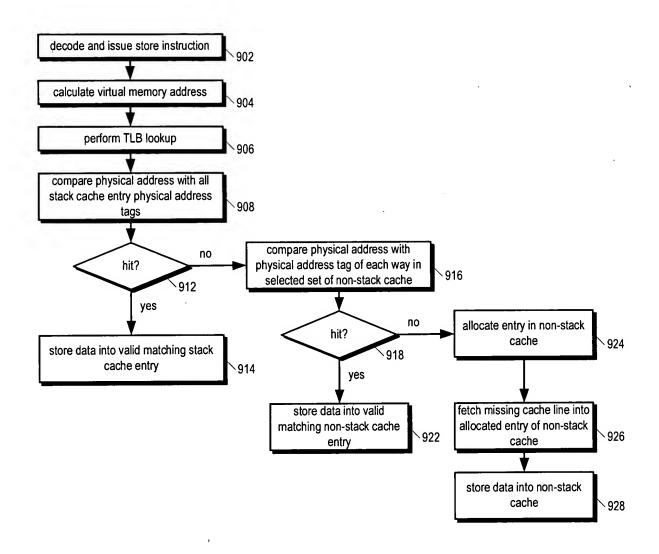
Load from Non-Stack Cache Operation



+

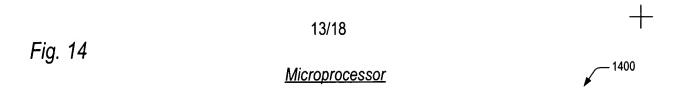
Fig. 9

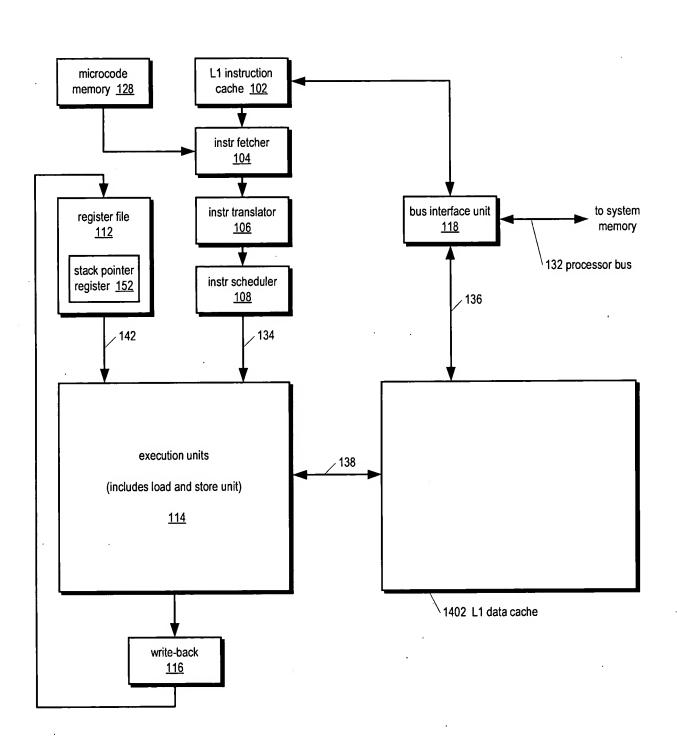
Store Operation



12/18 Fig. 10 Fast Pop from Stack Cache Timing clock cycle ==> 2 receive pop instruction request pop mux dword from cache line in top entry based on fp_offset pop calculate virtual address pop perform TLB lookup pop detect incorrect stack cache pop, based on physical address compare pop data available Fig. 11 Speculative Load from Stack Cache Timing 2 clock cycle ==> 3 4 receive load instruction request load calculate virtual address load load virtual tag compare and generate speculative load select from matches and valids perform TLB lookup load mux cache line based on speculative load select and mux dword based on PA[5:2] load load detect incorrect speculative load, based on physical address compare data available Fig. 12 Normal Load from Stack Cache Timing 1 2 clock cycle ==> receive load instruction request load calculate virtual address load perform TLB lookup load physical tag compare and generate normal load select from matches and valids load mux cache line based on normal load select and mux dword based on PA[5:2] load data available Fig. 13 Load from Non-Stack Cache Timing 5 6 clock cycle ==> 1 2 3 receive load instruction request load calculate virtual address load load perform TLB lookup row decode based on physical address index and array lookup load physical tag compare and generate way select based on matches and valids load mux cache line based on way select and mux dword based on PA[5:2] load

data available





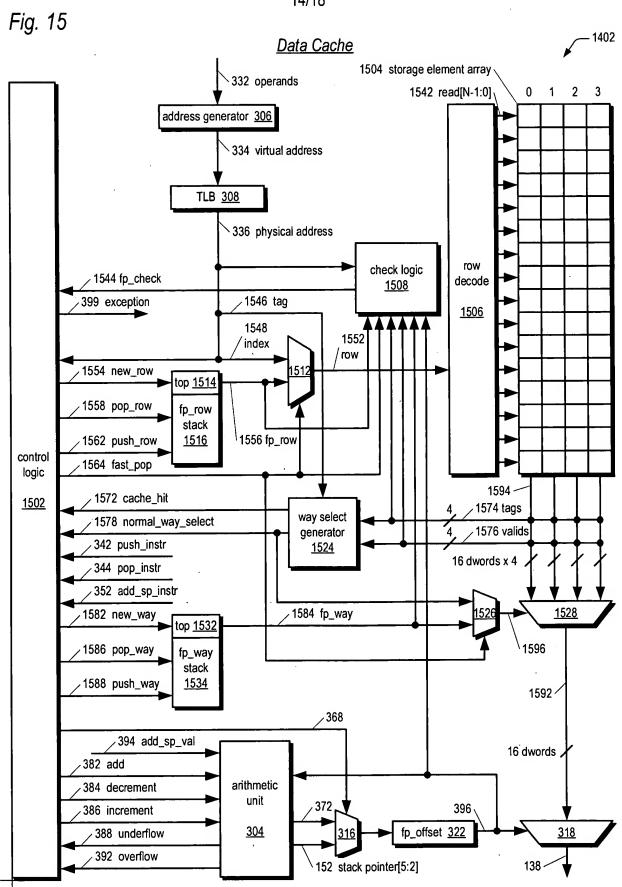
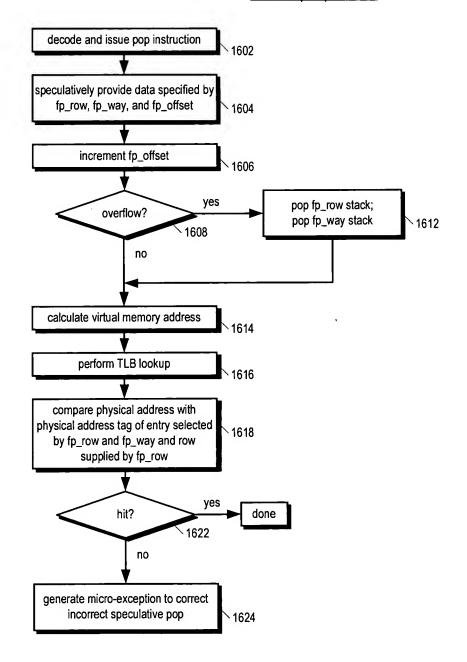


Fig. 16

Fast Pop Operation



l

Fig. 17

Push Operation

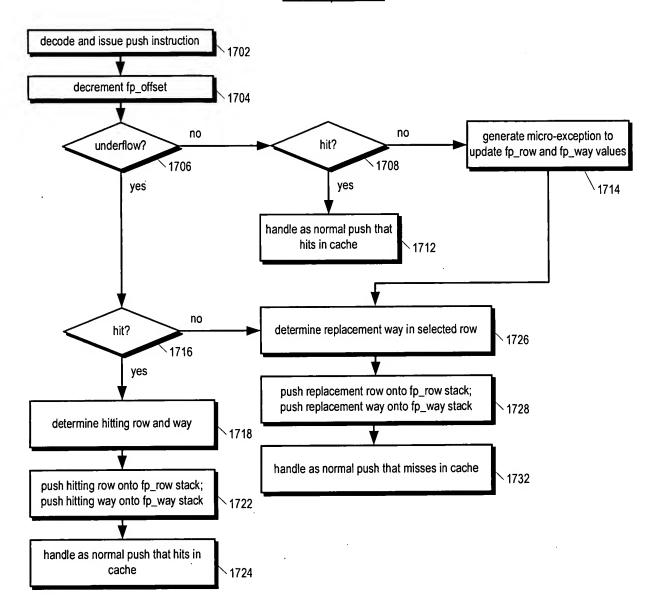
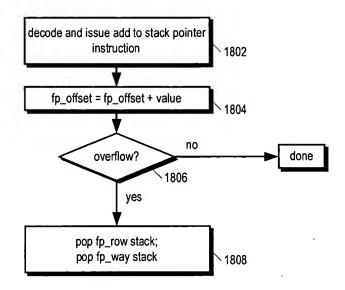


Fig. 18

Add to Stack Pointer Operation



_

Fig. 19

18/18 Fast Pop from Cache Timing

Tuot top not to the times				
clock cycle ==>	1	2	3	4
receive pop instruction request	рор		•	
row decode based on fp_row and array lookup		рор		
calculate virtual address		рор		
mux cache line based on fp_way and mux dword based on fp_offset		Ü	рор	
perform TLB lookup			рор	
detect incorrect speculative pop, based on physical address compare				рор
data available———				